САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Математико-Механический факультет
Кафедра системного программирования

Шалымов Роман Сергеевич

Разработка алгоритма определения авторства арабографического текста по статистическим данным

Курсовая работа

Научный руководитель:
проф. Граничин О.Н.

Санкт-Петербург
2015

Содержание

Введение

Стиль письма того или иного автора определяется тем, какие он подбирает слова, как строит предложения, какими оборотами пользуется и др. Когда необходимо автоматически определить автора данного текста используют методы, в научной литературе известные как **Authorship Attribution (AA)**. Такие задачи становятся особенно актуальными в последние два десятилетия благодаря стремительному росту текстовых документов в цифровой форме, которые необходимо организовать и обработать [1]. Задачи **AA** являются трудоемкими и эффективно используются в таких областях как верификация автора, выделение плагиата, профилирование автора (определение его возраста, пола, образования), выявление скрытых угроз (террористические угрозы) и др. Одной из наиболее важных проблем **AA** является выделение набора свойств (характеристик), которые будут использованы для определения автора.

В данной работе решается задача **AA** на основе статистического метода, описанного в [2]. Делается предположение, что автор – это псевдослучайный генератор символов, который может быть задан уникальным распределением. Тогда задача определения стиля сводится к задаче сравнения относительной независимости текстов.

Постановка задачи

Цель работы состоит в разработке алгоритма для автоматического распознавания авторства арабографического текста. При этом основным условием является устойчивость алгоритма к различным почеркам. Общая постановка задачи включает в себя следующие частные задачи:

1. Сегментация страницы документа на строки (**Text-line segmentation**)
2. Извлечение и обработка контура слов в строке (**Word Contour extraction**)
3. Кодирование контура специальной последовательностью символов (**Contour Hashing**)
4. Построение хэш-кода документа
5. Расчет корреляции между двумя генерированными последовательностями (**Kolmogorov-Smirnov Test**)

Сегментация документа на строки

Изначально предполагается, что тексты подаются в отсканированном виде и их необходимо перевести в формат, пригодный для дальнейшей структурированной обработки на компьютере. Для этого производится предобработка изображения, после которой отфильтровываются внешние шумы (случайные точки, помехи, которые могут возникнуть при сканировании и др.), а также адаптивно бинаризуется изображение, то есть переводится все в черно-белый формат. Далее для последующего анализа текста документа необходимо произвести сегментацию, состоящую в распознавании и выделении строк текста на изображении.

Документы, написанные на арабском языке, обладают специфичными характеристиками, такими как наличие множества диакритических точек и дополнительных знаков. Точки используются для различения некоторых арабских символов, имеющих те же основные формы. Другие арабские символы включают в себя специальные знаки, чтобы изменить оттенок письма. Когда используются диакритические символы (точки, специальные знаки), они появляются выше или ниже символов и пишутся как изолированные фигуры. Это делает разделяющую границу текстовой линии сложной. Действительно, диакритические символы могут генерировать лишние строки.

Арабская рукопись также очень курсивна и производит много надстрочных и подстрочных элементов, которые в целом касаются друг друга в различных текстовых строках, если линии расположены не очень далеко от друг друга. В большинстве случаев, пространство между текстовыми линиями наполнено множеством диакритических точек, знаков и дополнительных касающихся компонент, что делает сегментацию арабских документов трудоемкой задачей.

Несмотря на то, что в литературе было опубликовано множество методов для сегментации рукописных текстов для латинских и нелатинских скриптов, только немногие из них применимы к сегментации рукописных арабских документов.

Подход, основанный на методе притягивающих-отталкивающих сил представлен в [5] для извлечения несущих линий текста (baselines). Он заключается в итеративной адаптации y-координат предопределенного количества несущих линий. Пиксели изображения выступают в роли притягивающей силы, и уже извлеченные несущие линии действуют как силы отталкивания для строящихся новых базовых линий. Линии должны иметь одинаковую длину. Результатом является набор псевдо-базовых линий текста, каждая из которых проходит через тело слов. Метод применим к древним латинских текстам. Некоторые ошибки могут произойти, когда две соседние текстовые строки значительно касаются друг друга, и когда текстовые строки имеют разную длину. Этот метод не разделяет перекрывающиеся или касающиеся компоненты, так как ищется только несущая линия текста, и компоненты явно не присваиваются текстовым строкам.

Несколько подходов, описанных в исследованиях [7, 8] используют классический метод проекции профиля и ищут минимумы на профиле для сегментации изображения на текстовые линии. Однако, эти методы достаточно трудно распространить на рукописные документы, где текстовые линии связаны, рядом с друг с другом или искажены, потому что там нет четких минимумов. Первая адаптация метода профильной проекции состоит в разделении изображения документа на вертикальные полосы [10, 11]. Затем на каждой полосе находится профиль проекции, что делает метод терпимым к наклону и колебанию линии.

Эти методы [10, 11] могут обрабатывать перекрывающиеся и простые касающиеся компоненты, то есть компоненты, которые касаются не более двух текстовых строк. Тем не менее, они не расчитаны для документов с мульти-касающимися компонентами, то есть компонентами, которые касаются более двух текстовых линий.

Таким образом, исследовав разные алгоритмы, мы решили применить **Block Covering** алгоритм [9]. В отличие от стандартных методов, он способен обрабатывать тексты, написанные как горизонтально, так и под меняющимся углом (кривые линии). Он также устойчив к пересечению нескольких строк, что существенно повышает точность и качество системы.

Block Covering алгоритм

Этот метод основан на покрытии, без перекрытия, изображения документа прямогугольными блоками. Данный метод опирается на принцип выбора оптимальной ширины вертикальных полос и на статистический анализ высот блоков, а также анализ соседних блоков в вертикальных полосах.

Основная идея этого метода заключается в разделении входного изображения на несколько вертикальных полос с фиксированной шириной r. После этого, в каждой из полос все максимальные по включению блоки, содержащие только текстовые фрагменты, должны быть найдены. Нам необходимо выбрать все компоненты связности в каждой из вертикальной полос. При реализации мы просто итерируемся по строкам пикселей и рассматриваем сумму S из них. Если S обращается в нуль, то мы сравниваем ее с предыдущей суммой Sp. Если Sp больше, чем 0, то мы нашли конец текущего текстового фрагмента. Кроме того, если предыдущая сумма Sp равна 0 и текущая сумма S больше,чем 0, то мы нашли начало нового текстового фрагмента.

Эти блоки могут быть двух типов: блоки, охватывающие локальное пространство между линиями (пустые блоки) и блоки, охватывающие текстовые фрагменты (не пустые блоки). Ширина всех блоков фиксируется, но высота локально изменяется.

Большинство текстовых блоков включают неперекрывающиеся и не касающиеся компоненты. Таким образом мы можем извлечь из них некоторые статистические данные, а именно в нашем случае среднюю высоту текстовых линий. Аналогичным образом, высота большинства пустых блоков соответствует среднему локальному расстоянию между строками. Кроме того, локальные интервалы между строками трудно вычленить из множества связных компонент.

Block Covering алгоритм зависит только от ширины полос (strip) r, на которые мы разбиваем изображение. Поэтому нам необходимо также для каждого изображения научиться определять оптимальную ширину полос. Таким образом, он состоит из трех этапов:

1. Фрактальный анализ, приводящий к классификации документа на два класса: документы с маленьким межстрочным интервалом (TSD, tightly spaced document) и документы с большим межстрочным интервалом (WSD, widely spaced document) (r варьируется)

2. Статистический анализ высот блоков (r фиксировано)

3. Анализ блоков в соседних полосах (r фиксировано)

Документ, который в дальнейшем будет сегментирован, сначала анализируется и помечается в соответствии с подходящим классом (TSD или WSD). Для сегментации TSD документов, текстовые блоки подразделяются на три категории: небольшие текстовые блоки (им соответствуют диакритические знаки), средние блоки (составляют собой слова), крупные блоки (перекрывающиеся или касающиеся символы). Извлечение текстовых линий осуществляется после сегментации крупных блоков и состоит в присвоении каждому текстовому блоку соответсвующей текстовой линии. Следует отметить, что большие блоки могут быть разделены на более чем два блока при обработке мульти-касающихся строк. Для WSD документов текстовые блоки классифицируются на две категории: малые и средние блоки.

Классификация документа

В этом разделе описывается автоматический классификатор, который сортирует документы в две категории: WSD и TSD. При помощи фрактального анализа из изображения извлекаются два свойства. Эти две особенности подаются как входной вектор для **fuzzy C-Means** классификатора.

Фрактальный анализ

Размер объекта в d-мерном пространстве может быть посчитан как сумма размеров некоторых открытых множеств радиуса r и размера h(r).

Хаусдорф определяет в d-мерном пространстве меру множества F как

$$H_d(F) = \lim_{\epsilon \to 0} \inf_{r<\epsilon} \left\{ \sum h(r) : C(r) \right\}$$

где $C(r)$ является любым конечным покрытием F открытыми множествами размера $r < \epsilon$. При уменьшении радиуса r точность меры увеличивается, именно поэтому лучше брать предел при ϵ стремящимся к 0.

Хаусдорф показывает, что для любого множества F в d-мерном пространстве существует величина D_H (Хаусдорфова размерность) такая, что

$$H_d(F) = \begin{cases} \infty, & \text{если } d < D_H \\ 0, & \text{если } d > D_H \end{cases}$$

D_H известна как Хаусдорфова размерность, H_D называется Хаусдорфовой мерой, когда $d = D_H$.

В отличие от топологической размерности, эта размерность может быть не целой.

Бенуа Мандельброт в 1983 году ввел такое понятие как фрактальная размерность: Множество A в Евклидовом n-мерном пространстве называется самоподобным, если оно является объединением N дизъюнктных (не пересекающихся) копий самого себя, сжатых по каждой координате в r раз.

Фрактальная размерность множества:

$$D = \frac{logN(r)}{log(\frac{1}{r})}$$

Было показано, что при $r \to 0$ D стремится к Хаусдорфовой размерности.

Фрактальная геометрия может дать возможность квалифицировать рукописные тексты в зависимости от сложности графики. Однако, несмотря на то, что определение фрактальной размерности является достаточно простым, трудно оценить фрактальную размерность непосредственно из изображения.

Block Counting метод

В случае письма, фрактальная размерность – это мера того, каким образом письмо располагается в 2D геометрическом пространстве. Открытым множеством в данном случае выступает непустой блок с текстом; размер открытого множества равняется ширине вертикальных полос(strip), на которые разбивалось изначальное изображение, то есть $r = 1/vsn$, где vsn равняется числу вертикальных полос, однако высота варьируется и адаптируется под размер фрагмента.

Согласно теории Хаусдорфа, так как r является единственным существенным параметром, определяющим размер открытого множества, упрощенная мера Хаусдорфа для block counting метода выглядит следующим образом:

$$H_B = N_B(r) * (r)^D$$

где N_B – общее количество блоков ширины r, покрывающих весь текст.
Взяв логарифм от обоих частей, получим:

$$\log N_B = \log H_B + D * \log(1/r)$$

Если D интерпретировать как наклон прямой линии, а $\log H_B$ как начальную ординату, то линейность полученного графика связано с самоподобием фигуры.

Рассмотрим первое изображение, изображенное на рис.1. Оно содержит десять прямых горизонтальных линий, которые схематически изображают текстовые линии. Любая вертикальная полоса, вырезанная из всего изображения, будет самоподобна целому изображению. Применение формулы (1) дает $D = 1$ и $H = 10$. Параметр H отображает суммарное количество горизонтальных линий. Рисунок 1 показывает зависимость $\log N_B$ от $\log(1/r)$.

Изображение на рисунке 2 состоит из десяти горизонтальных линий с дополнительными бородками переменной высоты. Эти бородки имитируют перекрытие и соединение соседних полос текста, схематизируя запутанность линий текста в реальном документе. Свойство самоподобия не наблюдается при больших значениях r (Рис. 2). Мы наблюдаем нелинейное поведение графика. H имеет более низкое значение и больше не представляет собой количество полос. Линейность графика появляется, когда $r < r_0$ (в данном случае, $r_0 = 1/20$). Таким образом, свойство самоподобия достигается в этой области. H равно 10, что соответсвует числу горизонтальных линий на изображении.

Анализ предыдущих случаев показывает, что для нерегулярного случая (перекрытие или касание линий) фрактальная размерность D намного больше 1, и H отлична от количества линий текста (равного 10 в предыдущих примерах). В обычном случае, фрактальная размерность D равна 1, и H эквивалентна общему числу текстовых линий.

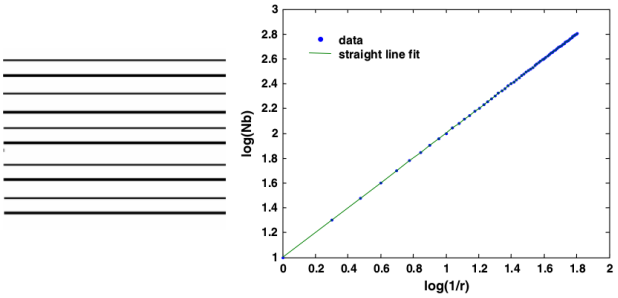


Рис. 1

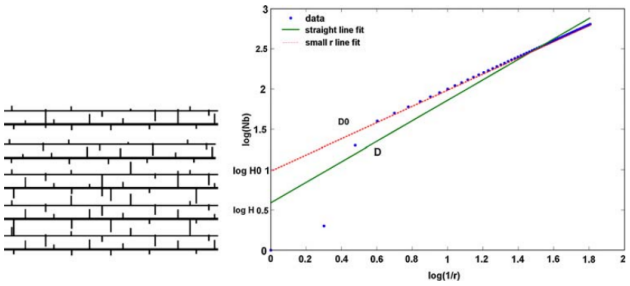


Рис. 2

В обычном и нерегулярном случае при малых значениях r наблюдается выравнивание точек, которое определяет автоматическую прямую линию симметрии, фрактальная размерность которой (угол наклона) обозначена D_0 и равна 1. Начальная ордината обозначена $logH_0$. H_0 может трактоваться как суммарное количество текстовых горизонтальных линий (даже перекрывающихся и пересекающихся), поскольку $D_0 = 1$ – это топологическая размерность линий.

В необычном случае, прямая линия, полученная линейной регрессией по всем точкам, имеет угол наклона D, отличный от D_0, и начальную ординату $\log H$, отличную от $\log H_0$. D является средней фрактальной размерностью и можно рассматривать как среднее количество сепарабельных линий, потому что H теоретически равно числу блоков, когда нет разрезания исходного изображения на несколько вертикальных полос (r = 1). В регулярном случае, усредненная прямая линия по всем точкам сливается с автоматической прямой линией симметрии. (D = D0, H = H0). Таким образом, можно выделить два класса документов (**TSD**, tightly spaced documents, плотно расположенные документы, и **WSD**, widely spaced documents, широко расположенные документы) с особенностями D/D_0 и $\log H_0/H$ (обозначается $\delta \log H$ в дальнейшем). Эти две особенности не зависят от размерности, то есть они не зависят от размера изображения. Этот метод используемый для искусственных изображений может быть примененен к настоящим изображениям, если график зависимости $\log Nb$ от $\log(1/r)$ показывает те же свойства.

При анализе изображений настоящих документов, когда все текстовые строки отделимы вертикальной проекцией, можно заметить, что линия аппроксимации имеет значение угла наклона очень близкое к 1, и что все особые точки почти лежат на этой линии (рис. 3). Когда линии касаются или перекрываются (рис. 4), D намного больше 1 (здесь D = 1,77), и зона линейности наблюдается при малых значениях и $\delta \log H$ значительно отлично от нуля (здесь $\delta \log H = 0,73$).

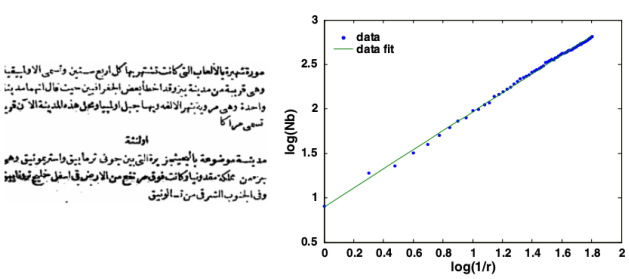


Рис. 3

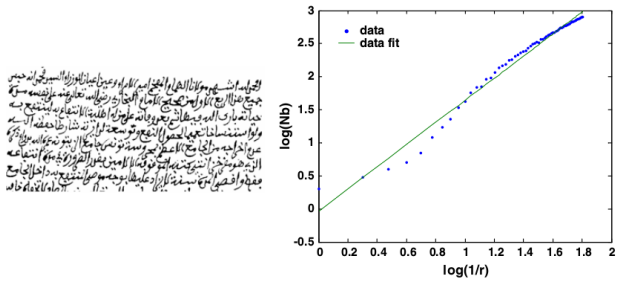


Рис. 4

В реальных случаях, D измеряет степень взаимопересечения текстовых линий: чем менее сепарабельны линии при помощи вертикальной проекции, тем больше фрактальная размерность. В самом деле разрезание на тонкие полоски может выявить новые блоки, которые были замаскированы при проекции в более широких полосах. Когда вертикальные полоски достаточно тонкие, текстовые линии становятся сепарабельными внутри полосы, и начинает появляться свойство самоподобия. Как результат, в этой зоне наблюдается линейность с углом наклона близким к 1.

График для реальных документов имеет те же аспекты, что и для схематических изображений. Таким образом, два предыдущих свойства также являются подходящими для обработки изображений настоящих документов.

Обучение и классификация

Вектор особенностей $(\delta \log H, D)$ является входным параметром для fuzzy C-means классификатора, который классифицирует тренировочные данные на два класса (WSD или TSD).

Эти два класса, WSD и TSD, определяются из тренировочного множества, и они представлены статистической моделью Махаланобиса. Действительно, распределение точек гораздо более компактно для класса WSD, чем для класса TSD. Расстояние Махаланобиса учитывает дисперсию класса при помощи ковариационной матрицы. Класс S_i представлен средним вектором X_i и ковариационной матрицей R_i. Расстояние документа X до класса S_i определяется формулой:

$$d(X, S_i) = \sqrt{(X - X_i)^T R_i^{-1} (X - X_i)}$$

Сегментация строк

На предыдущей стадии документ был помечен как WSD или TSD. На данном этапе также применяется Block Covering метод. Тем не менее, покрывающие блоки обрабатываются по-разному: ранее считалось только их общее количество. Сейчас блоки статистически классифицируются в зависимости от их высоты. Количество классов для блоков различается в соответствии с классом документа: три класса для TSD документа и два класса для WSD документа. На этой стадии автоматически находится оптимальное покрытие изображения блоками. Маленькие блоки соответствуют диакритическим знакам. Высота средних блоков соответствует высоте текстовых линий и фигур слов. Большие блоки получаются в результате слияния нескольких текстовых блоков, которые перекрываются или касаются друг друга. Такое предположение о происхождении высот блоков имеет место, только когда размер шрифта довольно однороден. Класс больших блоков включен только в TSD документы. Большие блоки далее сегментируются на несколько средних блоков. После этого документ, включающий только маленькие и средние блоки, сегментируется на текстовые линии при помощи процесса группирования блоков соседних вертикальных полос.

Классификация блоков TSD документов

TSD документ включает три вида блоков. Оптимальное покрытие блоками должно быть найдено для каждого изображения документа. Наилучшее покрытие блоками дает наилучшую классификацию. В качестве метода классификации используется K-Means алгоритм в одномерном случае. Слишком большое количество вертикальных полос приводит к большому количеству средних блоков, в то время как слишком малое количество полос производит слишком много

больших блоков. Таким образом, очень важно, чтобы было определено оптимальное количество полос. Критерий оптимизации должен способствовать высокой плотности внутри кластера и низкой между-кластерной плотности. В нашем случае, число классов фиксировано (тремя рассматриваемыми классами являются: маленький, средний и большой), но данные различаются (высота блока и число блоков) в зависимости от числа вертикальных полос. Количество вертикальных полос ($vsn = 1/r$) является настраиваемым параметром. Критерий должен дать высокую меру разделения кластеров, а также высокую внутрикластеровую плотность. Таким образом, можно максимизировать их произведение.

Критерий качества кластеризации блоков CDbw(Composing Density between and with clusters), зависящий от количества вертикальных полос vsn, определен как:
$CDbw(vsn) = intraDen(vsn) * sep(vsn)$.

Оптимальное количество вертикальных полос, обозначаемое ovsn, равно тому количеству полос, которое максимизирует критерий CDbw(vsn). Качество кластеризации внутри класса обозначено как $intraDen$. Мера разрозненности кластеров обозначена как $sep(vsn)$. Теперь мы вычисляем каждое из множителей:
Для заданного количества вертикальных полос $vsn = 1/r$, пусть $V_i = \{v_{i1}, v_{i2}, ..., v_{in_i}\}$ будет множество блоков i-ого класса, и n_i будет количеством блоков в этом классе. Стандартная девиация i-ого класса $stddev(i)$ определена как:

$$stddev(i) = \sqrt{\sum_{k=1}^{n_i} \frac{(h_{ik} - m_i)^2}{n_i - 1}}$$

где h_{ik} является высотой k-ого блока i-ого класса.
Средняя стандартная девиация $stddev$ определяется следующим образом:

$$stddev = \sqrt{\sum_{i=1}^{3} \frac{||stddev(i)||^2}{3}}$$

Качество кластеризации внутри класса, обозначенное $intraDen$ определяется как:

$$intraDen(vsn) = \frac{1}{3} \sum_{i=1}^{3} \sum_{j=1}^{n_i} density(v_{ij})$$

где $density(v_{ij})$ определено как:

$$density(v_{ij}) = \sum_{l=1}^{n_i} f(v_{il}, v_{ij})$$

и $f(v_{il}, v_{ij})$ опредяется следующим образом:

$$f(v_{il}, v_{ij}) = \begin{cases} 1, & \text{если } ||h_{il} - h_{ij}|| \leq stddev \\ 0, & \text{иначе} \end{cases}$$

Межклассовая плотность $InterDen$ определена как количество блоков, находящихся в близкой окрестности нескольких классов. Эта плотность должна быть низкой. Она определена как:

$$InterDen(vsn) = \sum_{i=1}^{3} \sum_{\substack{j=1 \\ i \neq j}}^{3} \frac{||m_i - m_j||}{||stddev(i) + stddev(j)||} \times density(u_{ij})$$

u_{ij} является виртуальным блоком высоты $h_{ij} = (m_i + m_j)/2$
$density(u_{ij})$ определена как:

$$density(u_{ij}) = \sum_{k=1}^{n_i+n_j} f(v_k, u_{ij})$$

где v_k принадлежит объединенному множеству блоков классов i и j.

$$f(v_k, u_{ij}) = \begin{cases} 1, & \text{если } ||h_k - h_{ij}|| \leq (||stddev(i)|| + ||stddev(j)||)/2 \\ 0, & \text{иначе} \end{cases}$$

Тогда мера разрозненности кластеров определяется как:

$$sep(vsn) = \sum_{i=1}^{3} \sum_{\substack{j=1 \\ j \neq i}}^{3} \frac{||m_i - m_j||}{1 + InterDen}$$

Рисунок 6 показывает изменение критерия CDbw для различных значений vsn при разделении WSD документа (рис. 5) на вертикальные полосы. Оптимальное значение ovsn в данном случае равно одиннадцати полосам.


ویسئلونک عن الروح (پ۱۵)
اور سوال کرتے ہیں تجھ سے روح کی نسبت
یعنی روح انسانی کیا چیز ہے۔ اس کی ماہیت و حقیقت کیا ہے؟
یہ سوال صحیحین کی روایت کے موافق یہود مدینہ نے آنحضرت صلی اللہ علیہ وسلم
کے آزمانے کو کیا تھا اور "سیر" کی روایت سے معلوم ہوتا ہے کہ مکہ میں "قریش"
نے "یہود" سے یہ سوال کیا تھا۔ اسی لئے آیت کے "مکی" اور "مدنی" ہونے
میں اختلاف ہے۔ ممکن ہے کہ نزول مکرر ہوا ہو، واللہ اعلم۔ یہاں اس سوال
کے درج کرنے سے غالباً یہ مقصود ہوگا کہ جن چیزوں کے سمجھنے کی ان لوگوں کو
ضرورت ہے۔ اُدھر سے تو اعراض کرتے ہیں اور غیر ضروری مسائل میں از راہ
تعنت و عناد جھگڑتے رہتے ہیں۔ ضرورت اس کی تھی کہ وحی قرآنی کی روح سے
باطنی زندگی حاصل کرتے اور اس نسخہ شفا سے فائدہ اُٹھاتے۔
وَكَذٰلِكَ اَوْحَيْنَآ اِلَيْكَ رُوْحًا مِّنْ اَمْرِنَا (شوریٰ رکوع ۵)
يُنَزِّلُ الْمَلٰٓئِكَةَ بِالرُّوْحِ مِنْ اَمْرِهٖ عَلٰى مَنْ يَّشَآءُ مِنْ عِبَادِهٖٓ (نحل رکوع ۱)
مگر انہیں دور از کار اور معاندانہ بحثوں سے فرصت کہاں؟
روح کیا ہے۔ جوہر ہے یا عرض؟ مادی ہے یا مجرد؟ بسیط ہے یا
مرکب؟ اس قسم کے نامفید اور بے ضرورت مسائل کے سمجھنے پر نجات موقوف
نہیں۔ نہ یہ بحثیں انبیاء کے فرائض تبلیغ میں داخل ہیں۔ بڑے بڑے حکماء اور فلاسفر
آج تک خود "مادہ" کی حقیقت پر مطلع نہ ہو سکے۔ روح جو بہر حال مادہ سے کہیں

Рис. 5

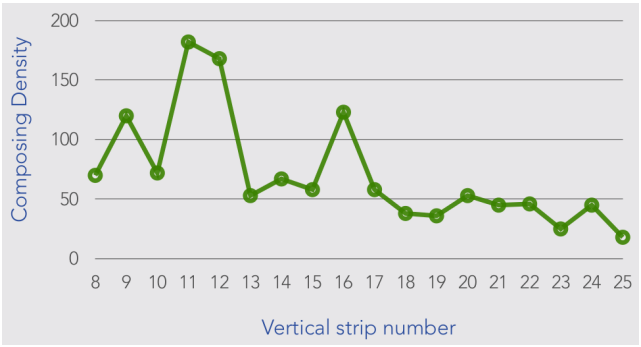


Рис. 6

Сегментация больших блоков

Большие блоки получаются в результате перекрытия или касания нескольких текстовых линий. Сегментация состоит в разделении большого блока на n средних блоков одинаковой высоты h, разделенных n - 1 пустыми блоками высоты e, где величины n,h и e адаптированы для каждого большого блока. Каждая тройка (n, h, e) может синтезировать большой блок высоты Hs. Таким образом, Hs равна $n * h + (n - 1) * e$.

Для больших блоков высоты Hs, тройка (n, h, e) выбирается так, чтобы наилучшим образом заполнить блок:

$$(h, n, e) = \arg_{(h,n,e)}(\min ||Hs - (n * h + (n - 1) * e)||)$$

Оценочное количество строк n должно удовлетворять $n \geq 2$.

Высота h для каждого большого блока выбирается из интервала $[mean(h_1), mean(h_1 + stddev(h_1))]$, где h_1 – случайная величина высоты среднего блока, что является статистически наиболее вероятным для довольно крупного среднего блока. $mean(h)$ и $stddev(h)$ – первые два статистических момента высот класса средних блоков.

Множество межстрочных пустых блоков является подмножеством всех пустых блоков. Все пустые блоки не обязательно принадлежат классу межстрочных блоков. Действительно, пустые блоки могут образоваться из-за: абзаца с отступом, неполной строки, отступов в начале или конце текста. Межстрочным блокам соответствуют наименьшие пустые блоки. Для разделения множества пустых блоков на два класса (межстрочные, не межстрочные) производится K-means кластеризация. Класс с наименьшей средней высотой соответствует межстрочным блокам. Сначала вычисляются первые два статистических момента $mean(e)$ и $stddev(e)$ на множестве межстрочных блоков, чтобы определить допустимый интервал для e: $[\sup(1, mean(e) - stddev(e)), mean(e)]$. Если несколько троек удовлетворяет этому условию, выбирается тройка с максимальной высотой h и минимальным значением e.

Сегментация WSD документов

Документы, классифицированные как WSD, не должны включать в себя большие блоки. Следовательно, мы могли бы разделить их только на одну вертикальную полосу. Тем не менее, метод классической вертикальной проекции недостаточно эффективен для сегментации документа на текстовые строки, поскольку диакритические знаки будут приводить к ложным линиям. Также ложное определение линий может произойти, если документы содержат небольшое количество касающихся или перекрывающихся компонент. Поэтому мы проводим все те же, шаги, что и для TSD документа, только в K-Means кластеризации классифицируем блоки на два класса вместо трех. Результирующие блоки с маленькой высотой соответствуют диакритическим знакам, средние блоки содержат основной контекст письма.

Построение текстовых линий

Для построения текстовых линий используется метод Neighboring Analysis. Его целью заключается в присваивании каждому блоку определенной текстовой линии. Этот процесс итеративный, и на каждой итерации обрабатываются пары соседних вертикальных полос (c_j и $_{j+1}$) слева направо. Все блоки в полосе c_j, которые принадлежат определенной конфигурации, помечаются в конце каждой итерации. Далее процесс итерируется по паре полос (c_{j+1}, c_{j+2}) пока не достигнет пары (c_{N-1}, c_N). Затем второй процесс происходит справа налево, начиная с пары (c_N, c_{N-1}) до (c_2, c_1) для того, чтобы промаркировать оставшиеся немаркированные блоки.

Существуют три конфигурации A-C для группировки двух блоков соседних полос в одной текстовой линии. Эти конфигурации основаны на вертикальном положении двух блоков относительно друг друга (рис. 7). Пусть $B_{J,K}$ будет K-ый блок в J-ой полосе, который мы хотим проанализировать. Когда блоки находятся в конфигурации A, они никогда не группируются. Группировка происходит только для конфигураций B и C. Но конфигурации B подразделяется на две конфигурации B1 и B2, когда блок в одной полосе имеет более одного соседа в соседней полосе (Рис. 8). Нам необходимо выбрать один из нескольких вариантов назначения текущему блоку текстовой линии соседних блоков. Наилучшее назначение зависит от того, маркированы ли уже другие соседние блоки или нет, а также от уровня перекрытия соседнего блока с текущим, который мы хотим промаркировать.

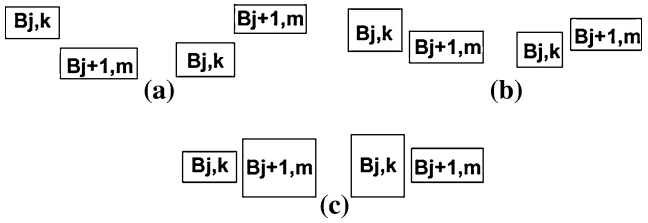


Рис. 7

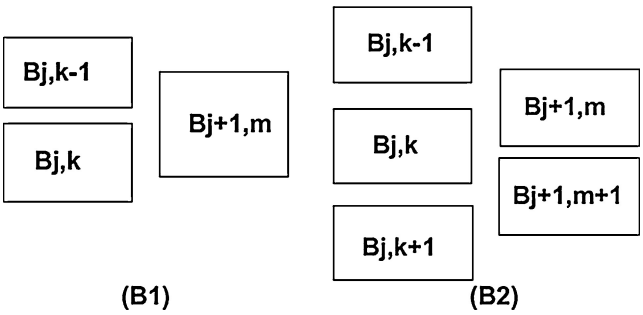


Рис. 8

Сегментация TSD документа на рис. 9а показана на рис. 9б. Две соседние линии показаны разным цветом. Единственные дефекты сегментации могут появиться только из-за сегментации больших блоков: разделяющие линии блока проходят через символы, и мелкие детали символа присваиваются соседней линии. Но для WSD документов (Рис. 9c, 9e) такие дефекты отсутсвуют, поскольку отстутствуют большие блоки, и нет разделяющих линий.

Эксперименты проводились на 215 страницах WSD документов и 65 страниц TSD документов, качество сегментации WSD документов составило 97 процентов, для TSD – 85 процентов.

Рис. 9

Кодирование документа

Мы предполагаем, что исходный документ уже был сегментирован на текстовые строки. Прежде, чем перейти к стадии получения кодовой последовательности документа, нам необходимо качественно извлечь контуры слов в каждой текстовой линии. Качество распознавания рукописного текста высоко зависит от качества входных изображений. Мы работаем на отсканированных изображениях, поэтому проводится их предварительная обработка для более качественного извлечения контуров слов в дальнейшем.

Перед извлечением контуров слов мы увеличиваем изображение (image magnification) текстовой строки, что делает важные особенности контуров фигур более заметными. Случайное расщепление слова может произойти как при письме, так и при сканировании. Чтобы уменьшить этот эффект, мы применяем к изображению оператор morphological dilation. Теперь для увеличения качества нахождения контуров предварительно применяется Canny edge detection оператор, и после этого извлекаются все контуры из изображения.

Представление контура

После предобработки контур слова доступен как массив точек (x, y):

$$[(x_1, y_1), (x_2, y_2), ..., (x_N, y_N), (x_1, y_1)]$$

Иногда более удобно с вычислительной точки зрения представить контур в комплексном пространстве. В этом пространстве мы имеем дело с одномерным массивом комплексных чисел:

$$[z_1, z_2, ..., z_N, z_1], \text{ где } z_n = x_n + i \cdot y_n$$

Эта замкнутая кривая достаточно хорошо отражает структуру фигуры слова. Однако, поскольку контур построен с помощью boundary tracing, она недостаточно гладкая для дальнейшей обработки.

Сглаживающий (низкочастотный) фильтр решает данную проблему. Один из лучших способов сглаживания контура это использование дескрипторов Фурье. В этом методе вычисляется дискретное преобразование Фурье контура $z[n]$, и удаляются высокочастотные коэффициенты:

$$Z[u] = \sum_{n=0}^{N-1} z_n e^{-i2\pi un/N}$$

Модифицированная сглаженная кривая получается путем применения обратного преобразования Фурье оставшихся коэффициентов (первые P коэффициентов):

$$\tilde{z}_n = \frac{1}{P} \sum_{u=0}^{P-1} Z[u] e^{i2\pi un/N}$$

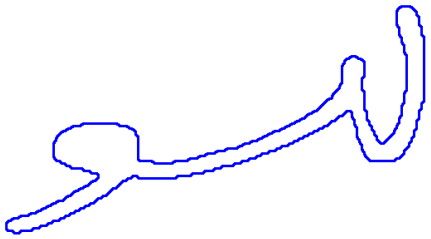
Наши эксперименты показывают, что достаточно брать 30 – 60 низкочастотных коэффициентов для получения сглаженной кривой приемлемого качества, отображающей все достаточные детали фигуры для последующих шагов.

Кодирование контура слова

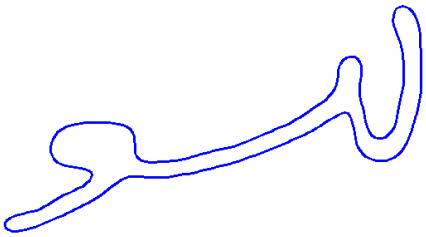
Теперь мы кодируем каждое слово специальной последовательностью символов (hash), основанной на особенностях фигуры контура слова, а именно на выпуклостях и вогнутостях контура.

Чтобы определить характеристические точки, каждый контур разбивается на верхнюю и нижнюю кривую. Чтобы получить разбиение, для начала мы находим точки контура с максимальным и минимальным значениями в направлении X. Если мы будем двигаться по контуру от точки с минимальным X к точке с максимальным X по часовой стрелке, то мы извлечем верхнюю кривую. Оставшаяся часть контура представляет собой нижнюю кривую. Затем мы находим пики и впадины взятием производной гладкого контура по x.

Теперь последовательность экстремумов кодируется как характеристическая строка. Каждая впадина кодируется как '1', если она находится на нижней кривой, и кодируется как '3', если она находится на верхней кривой. Аналогично, каждый пик кодируется как '2', если он находится на нижней кривой, и кодируется как '4', если он находится на верхней кривой. Также мы рассматриваем координаты точек экстремума: во-первых, значения координат точки x и y нормируются к [-1,1] и затем кодируются символами из ASCII диапазона



(a) Исходный контур

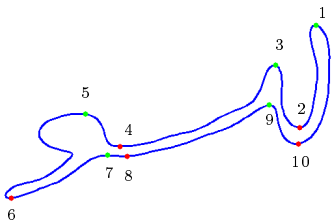


(b) Сглаженный контур

Рис. 11 Пример сглаживания контура слова

'a..zA..Z'. А именно, интервал [-1,1] делится на 52 подинтервала, каждому подинтервалу присваивается один символ. Если нормированная координата попадает в какой-то интервал, тогда координата кодируется соответствующим символом. Таким образом, каждая характеристическая точка представлена тремя аттрибутами: типом точки, закодированные нормированные координаты x и y. Рисунок 12 показывает пример кодирования контура. Строка построена по часовой стрелке. Стартовой точкой строки по соглашению является самая правая точка на верхнем контуре.

№	type	x	x code	y	y code
1	'4'	0.916	'W'	-1	'a'
2	'3'	0.815	'U'	0.185	'D'
3	'4'	0.667	'Q'	−0.54	'k'
4	'3'	−0.293	'r'	0.403	'J'
5	'4'	−0.506	'l'	0.026	'A'
6	'1'	−0.964	'a'	1	'Z'
7	'2'	−0.369	'p'	0.502	'M'
8	'1'	−0.249	's'	0.517	'M'
9	'2'	0.627	'P'	−0.08	'w'
10	'1'	0.807	'T'	0.374	'I'



Word Hash: 4Wa3UD4Qk3rJ4lA1aZ2pM1sM2Pw1TI

Рис.12: Пример кодирования контура

Построение хэш-кода документа

На этом этапе мы просто конкатенируем все хэши слов, которые находятся на одной текстовой линии, таким образом, получая хэш-код строки. Более формально, пусть $s = \{w_k\}_{k=1}^{n}$, где w_k – k-ое слово текстовой строки, отсчитывая с правой стороны. Тогда хэш строки есть

$$H(s) = \prod_{k=1}^{n} H(w_k),$$

где $H(w_k)$ – хэш-код слова w_k, n – число слов в текстовой линии и $*$ является операцией конкатенации строк.

Аналогичным образом, мы определяем хэш-код документа H_D, как последовательную конкатенацию хэш-кодов строк, то есть:

$$H_D = \prod_{i=1}^{N} H(s_i),$$

где N – количество текстовых линий документа.

Определение авторства

При определении авторства текстов мы выдвигаем гипотезу, что каждый автор является генератором псевдослучайный символов. Каждый документ, представленный в закодированном виде, является образцом из семейства с какой-то уникальной функцией распределения, присущей индивидуально каждому автору. Для сравнения двух закодированных документов мы применяем тест Колмогорова-Смирнова.

Тест Колмогорова-Смирнова (КС-тест) пытается определить, существенно ли различаются два набора данных. КС-тест имеет преимущество не делать никаких предположений о распределении данных.

One-sample тест

Этот тест Колмогорова-Смирнова используется для проверки того, что образец происходит из семейства с каким-либо известным распределением.

Пусть $F_0(x)$ – функция распределения предполагаемого распределения. Тогда $F(x)$ – эмпирическая функция распределения образца определена как:

$$F(x) = \frac{1}{n} \sum_{i=1}^{n} I_{\{X_i \leq x\}}$$

где I_A – это функция-индикатор, равная 1, если случилось событие A, и равная 0 иначе.

Теперь мы можем сравнить эту полученную функцию распределения F(x) с предполагаемой функцией распределения $F_0()$, чтобы измерить их сходство. Один из самых простых и интуитивно понятных измерений является наибольшее вертикальное расстояние D между функциями, то есть

$$D = \sup_t |F(t) - F_0(t)|$$

Эта дистанция D была предложена Колмогоровым и известна как *статистика критерия Колмогорова.*

Как мы уже упоминали ранее, мы хотим знать, принадлежит ли данный образец семейству с известным распределением. Формально говоря, мы заинтересованы в тестировании нулевой гипотезы формы:
$H_0 : F(x) = F_0(x) x - \infty + \infty$
$H_1 : F(x) \neq F_0(x) x$
Если D больше, чем $1 - \alpha$, то мы отклоняем H_0 на уровне точности α, иначе гипотеза H_0 считается принятой.

Two-sample тест

Пусть $X = X_1, X_2, ..., X_m$ и $Y = Y_1, Y_2, ..., Y_m$ – два независимых образца, чьи функции распределения F и G неизвестны. Аналогично one-sample тесту, two-sample тест состоит в проверке нулевой гипотезы
$H_0 : F(x) = G(x)$ для всех x
против альтернативной
$H_0 : F(x) \neq G(x)$ для некоторого x

Пусть $\tilde{F}(x)$ и $\tilde{G}(x)$ – эмпирические функции распределения образцов X и Y соответственно. Тогда статистика Колмогорова-Смирнова определена как

$$D = \sup_t |\tilde{F}(t) - \tilde{G}(t)|$$

Аналогично, если D больше $1 - \alpha$, то мы отклоняем H_0 на уровне точности α, иначе гипотеза H_0 считается принятой.

Таким образом, мы можем ввести метрику на множестве эмпирических функций распределения для закодированных документов.

Результаты

Для реализации системы был выбран язык программирования C++, а также использовалась библиотека OpenCV. В ходе работы был полностью реализован и протестирован алгоритм сегментации рукописного документа на строки. Также было реализовано извлечение и обработка контура слов в строке, кодирование контура специальной последовательностью символов и получение хэш-кода документа.

Эксперименты проводились на отсканированных рукописях Ибн Араби, а также его ученика, которые были предоставлены восточным факультетом Санкт-Петербургского государственного университета.

Дальнейшие направления работы

- Улучшение алгоритма сегментации документа на строки
- Аппробация теста Колмогорова-Смирнова на полученных хэш-кодах документов.
- Модернизация алгоритма кодирования контура слова

  Идея заключается в том, чтобы увеличивать или изменять извлекаемые из контура характеристические свойства для достижения более качественных показателей.

Список литературы

[1] E. Stamatatos, *A Survey of Modern Authorship Attribution Methods*, Journal of the American Society for Information Science and Technology, Vol. 60(3), 2009.

[2] P.Juola *Authorship attribution*, Foundations and Trends in Information Retrieval, Vol. 1, No. 3, 2006.

[3] Z. Volkovich, Z. Barzily, G.-W. Weber, D. Toledano-Kitai and R. Avros, *Re-sampling Approach for Cluster Model Selection*, Machine Learning, Vol. 85, 2011.

[4] J. H. Friedman, L. C. Rafsky *Multivariate generalizations of the Wolfowitz and Smirnov twosample tests*, Annals of Statistics, Vol. 7, 1979.

[5] R. Saabni, A. Asi, J. El-Sana, *Text line extraction for historical document images*, Pattern Recognition Letters, Vol. 35, 2014.

[6] K. Fouladi, B. N. Araabi, E. Kabir, *A fast and accurate contour-based method for writer-dependent offline handwritten Farsi/Arabic subwords recognition*, International Journal on Document Analysis and Recognition, Vol. 17, 2014.

[7] Khorsheed MS (2002) Off-Line Arabic character recognition—a review. Pattern Anal Appl 5:31–45

[8] Lorigo LM, Govindaraju V (2006) Off-line Arabic handwriting recognition—a survey. IEEE PAMI 28(5):712–724

[9] A. Zahour, B. Taconet, L. Likforman-Sulem, W. Boussellaa, *Overlapping and multi-touching text-line segmentation by Block Covering analysis*, Pattern Analysis and Applications, Vol. 12, 2009.

[10] Arivazhagan M, Srinivasan H, Srihari S (2007) A statistical approach to line segmentation in handwritten documents. In: Proceedings of Document Recognition and Retrieval XIV, IST&SPIE, San Jose

[11] Zahour A, Taconet B, Mercy P, Ramdane S (2001) Arabic hand- written text-line extraction. In: Proceedings of ICDAR'01, 10–13 Sept., Seattle, USA, pp 281–285